Methodology for Measuring the Complexity of Enterprise Information Systems

Ilja Holub

Department of Information Technology
Faculty of Informatics and Statistics
University of Economics, Prague
Czech Republic

ilja.holub@vse.cz

DOI: 10.20470/jsi.v7i3.260

Abstract: *The complexity of enterprise information systems is currently a challenge faced not only by IT professionals and project managers, but also by the users of such systems. Current methodologies and frameworks used to design and implement information systems do not specifically deal with the issue of their complexity and, apart from few exceptions, do not at all attempt to simplify the complexity. This article presents the author's own methodology for managing complexity, which can be used to complement any other methodology and which helps limit the growth of complexity. It introduces its own definition and metric of complexity, which it defines as the sum of entities of the individual UML models of the given system, which are selected according to the MMDIS methodology so as to consistently describe all relevant content dimensions of the system. The main objective is to propose a methodology to manage information system complexity and to verify it in practice on a real-life SAP implementation project.*

Key words: Complexity, methodology, metric, enterprise architecture, information strategy, information system, ERP, SAP, MMDIS.

1. Introduction

Complexity is a term that is today frequently used not only in specialized publications but also in methodologies and standards used to manage information systems and ICT projects. However, this term is rarely clearly defined. Intuitively, we understand the term of complexity as the ICT size, number of features and functions, and the considerable costs of implementing, operating and maintaining the same. From different viewpoints, we can identify specific aspects of ICT complexity: amount of data and information, complex processes, broad business organization structure, unfriendly user interfaces, sophisticated SW and HW – all these surely impact the entire organization, its staff, financial performance but also on one another. It is obvious that a business whose processes and the entire business architecture are complex will require a complex information system (IS) or even multiple information systems.

A good theoretical basis for describing and exploring complexity can be found in the mathematical theory of charts (Diestel, 2010)(Dehmer, 2010), which can be applied to some IS/ICT models to facilitate their quantification and subsequent comparison and, in some cases, even their simplification. Publications that come closest to the topic of business information system complexity deal with using metrics to measure complexity, such as information flow metrics, (Bansal and Negi, 2008) Halstead's method or functional point analysis (Phukan et al., 2005), or a simple number of source code lines (Geer, 2009). There are also publications dealing with the impacts of complexity on system security and maintainability. Their objective is to adapt the MATra framework to describe certain aspects of complexity (Mason and Cosh, 2008) and/or to point out the negative impacts of complexity on the business (Geer, 2009).

The topic of managing information system complexity is discussed, for example, by John Maeda in his book 'The Law of Simplicity', where he defines ten rules of simplicity. (Maeda, 2006) ERP systems are specifically discussed in 'A Metric for ERP Complexity' (Bansal and Negi, 2008), which counts process inputs and outputs and works with the concepts of internal and external module complexity, applying data flow complexity according to (Kafura and Henry, 1981). Best practices for simplicity of IS describes the book „Managing Complexity of Information Systems"(Lemberger and Morel, 2013).

From among Czech authors, we should definitely mention the works of authors Molnár (Molnár, 2001), Novotný (Novotný, 2007), Učeň (Učeň, 2001) and Maryška (Maryška, 2006), who present specific metrics that can be used to manage business information systems and ICT effectiveness, and Holub (Holub, 2012, 2011) , who defines complexity metrics of specific IS dimensions and causal relations between them.

In present-day methodologies used to manage ICT and information system projects the issue of complexity is rather insufficiently discussed.

Although some of them do mention the risks and costs arising from high complexity, none of them presents a method how to handle or even reduce complexity. The author has prepared a summary overview of the most frequently used methodologies in the following **Chyba! Nenalezen zdroj odkazů.**

Table 1 – Occurrences of the word "complex" in various methodologies; source: author

Methodology Title	Pages	Occurrence of the word "complex"	Occurrence of the word "simplify"	Number of solutions described
The Val IT Framework 2.0	116	6	1	0
TOGAF Version 9	778	74	11	5
Rational Unified Process	21	3	0	0
Prince 2	326	26	0	0
PMBOK	403	35	1	0
COBIT5	94	9	0	0

From among the methodologies listed here, TOGAF is the one that mentions complexity most often, and it is the only methodology that discusses this topic at least briefly. In its Section 1.2., it mentions a reduced IT infrastructure complexity as a prerequisite for reducing risks and improving return on investment. In Section 5.5.1, it states that a complex architecture is difficult to manage, and then in Section 5.5.4 it recommends splitting it in partial phases in such cases. A benefit of this methodology in respect of complexity can be found in Section 22 where it is stated that there is a risk of greater complexity stemming from the use of the SOA architecture. A new ADM method is presented as a way to develop new architecture, which is described in detail in Sections 18 and 19. The ADM is based on an iterative approach to optimization. It mentions that reducing business architecture complexity is a way to reduce costs; however, there is no method presented how to compare complexity and neither any metrics that could be used to measure business architecture complexity.

2. IHC Complexity Management Methodology

The objective of the IHC (Identify High Complexity) methodology is to reduce IS complexity or its dimensions at every stage of its lifecycle. As shown in the previous works published by the author (Holub, 2012, n.d.), it is easier to prevent complexity growth than to eliminate complexity from a system subsequently. One of the examples presented is the impossibility to remove a register entry from
a relational database if there are dependent tables using such an entry.

Therefore, the solution is to control all relevant dimensions at every stage of IS development, measure their complexity, identify and eliminate the unnecessary one in every artifact before it becomes an input for the creation and development of another one. The following procedure is also shown on Figure 1.

1. Select relevant dimensions and their models
2. Create models for all selected dimensions
3. Calculate the complexity
4. Simplify models
5. Calculate the finally complexity

2.1 Principles

The IHC methodology recommends to use following principles:

Simplify the requirement

Clearly define the objective, clients, outputs and stakeholders and analyze whether they present a potential for simplification. If the assignment contributes to greater system complexity, verify whether it is consistent with the IT strategy and overall business strategy. If not, try to reject the assignment.

Simplify all dimensions; if this cannot be done, do not change them

Identify elements in dimensions that can be removed or replaced with a lower number of other elements or entities. Consider whether complex dimensions need to be changed in a project or whether they can be left unchanged, and focus on those that truly relate to achieving the objective.

Implement only processes with a sufficient degree of maturity

If a process shows a low degree of maturity, which we can establish primarily during its modeling, that is difficult or even impossible in such a case, consider whether you are able to stabilize the process; if not, refuse to implement it or exclude it from the information system until it stabilizes itself.

Figure 1 – Complexity management procedure; source: author

2.2 Roles

This methodology is designed for all project stakeholders, whose task is to simplify inputs that they deliver to the project. Due to the high number of roles and their variants that are involved in IT projects, for the purpose of this methodology we will group them into three basic groups: Sponsor, Supplier and Operator

Client

The client is an entity that desires to support its processes by an information system. Be it a business that plans to implement a new information system, a department that introduces a new process or an employee who needs to generate an analytical report from an existing system, they always communicate it as a request to the Supplier.

The following key parameters are involved in terms of complexity management:

- List of stakeholders
- Process model and process inputs and outputs

- Maturity of the process that is supposed to support the solution
- Required implementation time (time to market)
- Expected budget
- Required lifetime of solution
- Data

Supplier

The Supplier is an entity responsible for implementing and, sometimes, also supporting and maintaining the solution. It can be the IT department of the company, contracted SW provider, system integrator or a project team consisting of external and internal specialists.

The following key parameters are involved in terms of complexity management:

- Number of ICT suppliers/providers
- Number of information systems
- Business process optimization proposal depending on functions of the existing IS
- Solution designs
- Documentation

Operator

The Operator is an entity that operates and maintains the information system and ensures the security and availability of the system or its part through its service life. The solution provider may also act as the Operator. It can as well be the company's IT department or an external service provider. The following parameters are key from the Operator's viewpoint:

- Costs of solution operation
- Risks
- Costs of changes and remedying faults
- Methodologies, documentation
- Security constraints

2.3 Definition and Calculation of Information System Complexity

In accordance with the MMDIS methodology, in order to describe system complexity, we will assign a UML model to each of its content dimensions, and then we will define system complexity as a sum of the complexities of each UML model of its content dimensions. We select those content dimensions that can be quantified. Therefore, we will specify the content dimension definitions so that they can be modeled using the UML language.

An entity in this methodology is understood as an element or connection. That means a vertex or an edge if we consider the chart as a model analogy. We will use the MMDIS content dimensions (Table 2), to which we will assign UML models commonly used to model business processes and information systems. (Řepa and Česká společnost pro systémovou integraci., 2006)

Table 2 - Selected content dimensions according to the MMDIS; source: author

Title	Abbreviation	Model
Functions/processes	PRO	Activity diagram/BPMN
Data/information	INF	Class Diagram
Organizational and legislative aspects	ORG	Org Chart Diagram
Labor, social and ethical aspects – human resource aspects	PRA	Stakeholder matrix
Software	SW	Component diagram
Hardware	HW	Deployment diagram
User interface	UR	GUI model
Security	BE	Risk management plan
Financial aspects	EKO	Activity Based Costing

2.4 Dimensions and Modeling

To measure information system complexity, we need to select suitable content dimensions and modeling tools, which we then use to create their model. Depending on the information system lifecycle stage and type, some dimensions may be more important and others less important. In this step, we also need to decide whether the metrics used will be the number of elements, number of connections between them or the number of all entities.

In order to measure complexity, we need to set up the metrics parameters as shown in **Table 3**.

Table 3 – Set up metrics parameters to measure dimension complexity; source: author

	Dimension title
Complexity	Precisely define the measured dimension
Metrics	Number of links and/or elements
Model	Describe the model or state that it is a simple list of elements
Tool	A tool used to model or establish complexity
Granularity (element)	Describe the element
Granularity (link)	Indicate whether the link orientation is considered

The next, more time-consuming phase is modeling each of the dimensions. The selection of the modeling tool and the relevant methodologies and granularities do not play a role here. It is important to observe this for all systems that we are going to compare. It is also important to verify the consistency of resulting models. (Řepa and Česká společnost pro systémovou integraci., 2006) Preparing models is not purposeless: apart from inputs for complexity calculations, the model serves as a basis for designing an information system solution and, in particular, a suitable model forms a prerequisite to simplifying any dimension.

When measuring system complexity, the first step is to select relevant content dimensions D and the IS lifecycle phase *f*, using which we wish to measure complexity according to the MMDIS methodology. Let us designate their number as *d*. For dimension *i* in phase *f* we can define complexity $C_f(D_i)$ as the number of entities (elements and/or links) of its UML model.

Let us define system complexity C(S) as a sum of the complexities of each of its dimensions $C(D_i)$. Thus, complexity C of system S in phase *f* is the number of entities (elements and/or links) in the models of each of its content dimensions (D) selected in the given phase; *d* shall be the total number of dimensions considered.

$$C_f(S) = \sum_{i=1}^{d} C_f(D_i)\,, \qquad (1)$$

This formula (1) allows us to quantify the complexity of a system or its part, and in practice it can be used to compare various systems or solution alternatives that meet the given requirements, yet there is no objective criterion why one of them should be selected.

At the same time, let us introduce the concepts of accidental (undesirable) and essential (necessary, justified, optimum) complexity.

A system is excessively complex if it contains elements or links that are not necessary for its required functionality.

On the other hand, essential complexity means the minimum possible system configuration that will ensure the required functionality.

2.5 Simplification

Models of specific dimensions form the basis for simplification as they do for complexity measurement.

Solution Integration Method

Methods of integrating new information systems, ICT components or new functionalities in the existing architecture can be divided into the following three categories:

- Immediate integration
- A standalone solution planned to be integrated later on
- A standalone temporary solution planned to be discarded after some time

From the viewpoint of retaining an integrated and consistent architecture, integration should be the option of choice; however, there may be reasons that do not always allow it. Integration requires cooperation with specialists in the existing system components, which means higher costs of and more time for implementation, testing and commissioning. In larger enterprises, another requirement is the time consistency with the release schedule. If a request for implementation is linked to the budget or schedule, and for these reasons it does not make integration possible, it is necessary to implement a standalone solution. Gartner introduces the concept of 'bimodal IT', which recommends that some requests be implemented directly as standalone solutions depending on the solution nature, as **Figure 2** shows.

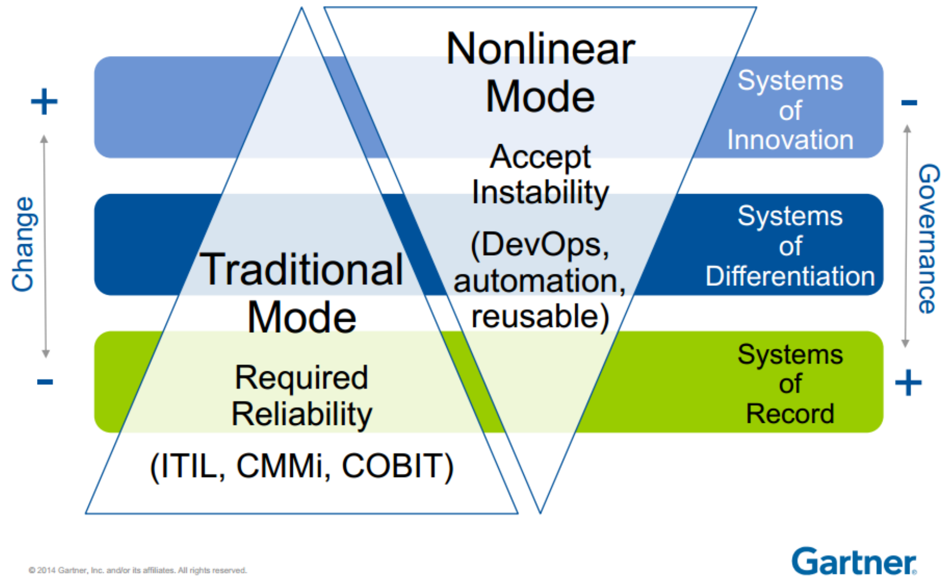

Figure 2 – Bimodal solution variants according to Gartner; source: Gartner (Gartner, 2015)

If a decision is made to implement a standalone solution, it needs to be integrated or discarded within a pre-defined period of time. Otherwise, non-integrated solutions will begin to pile up in the enterprise, which will disrupt the overall architecture, increase its complexity, and make the implementation of every subsequent change much harder.

Business Process Complexity

For the purpose of this methodology, we define business process complexity as the number of activities in the process model. Therefore, we start with a process model of a certain granularity that describes the existing business processes. The principles and procedures of process modeling exceed the scope of this article, and there are sufficient tools and literature available on this topic.(Repa and Bruckner, 2015) This methodology does not include instructions how to reduce business process complexity or how to optimize such processes, but it views process complexity as an input parameter when determining information system complexity.

Figure 8 shows a model of a sample sales process in a business, created using ARIS Express. Model granularity is of importance here: it is necessary to show every activity that one user performs in one system at a given moment. For our purposes, the concept of user here also includes the system user, i.e., activities performed automatically by the information system.

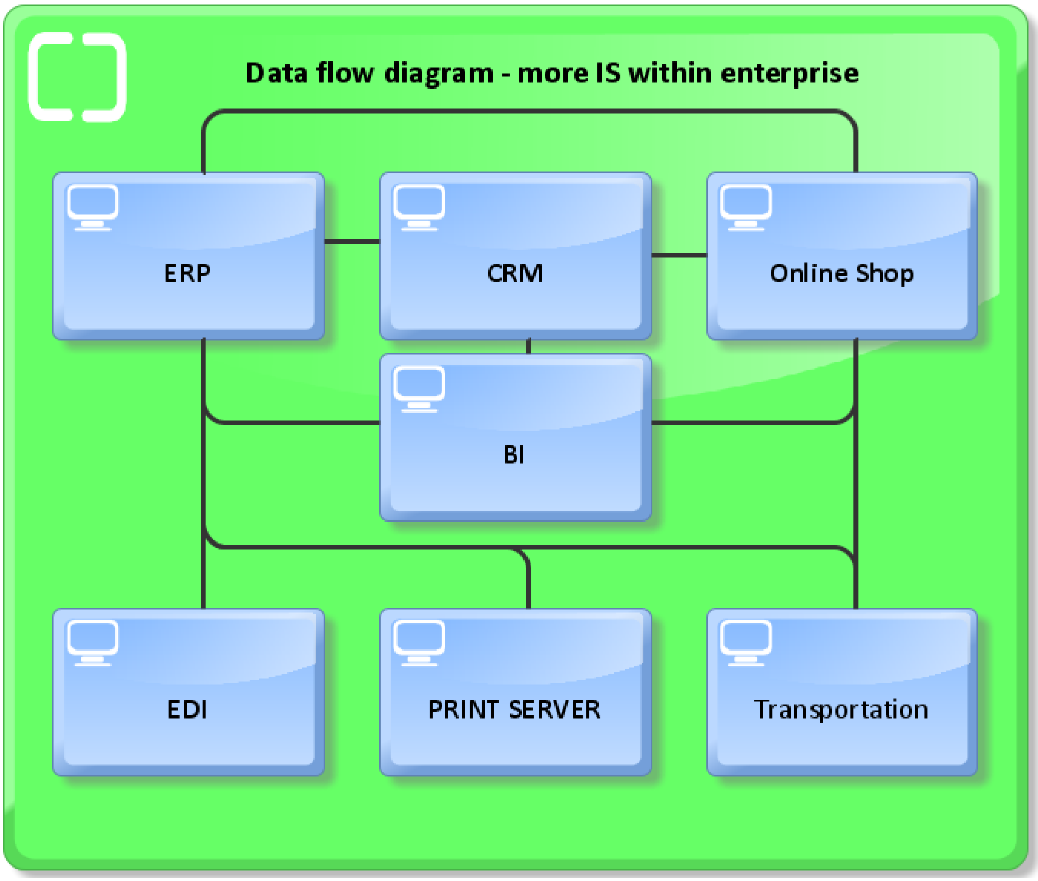

Figure 3 - Customer data flows between business information systems; source: author

Data Architecture

The data dimension has a major influence on information system complexity. When designing a system, we will start with using a process model and identify the main data entities in the enterprise, such as the customer, material, supplier etc., and then we prepare a DFD[1] diagram.(Řepa and Česká společnost pro systémovou integraci., 2006) The DFD diagram shows each of the data storage elements and the data flows between them. The objective is for every entity to have just one storage and for the data flows to be as simple as possible, preferably within a single system and single database. Usually, an information system is not designed as a greenfield project; rather, a business already uses some systems, which will have to be replaced either wholly or partially. In the next step, we therefore need to decide on the position of each of the storage elements between the systems that will be operated in the enterprise during the years to come. According to Principles 1 and 2, we will try to place as many data storage elements as possible in one of the systems and then create a system architecture model or the 'landscape'. We thus use the following models:

- Process model
- Data model
- Landscape

In the case of multiple systems and multiple data storage elements for one object, the number of links increases. A typical situation is illustrated by Figure 3.

According to this methodology, data architecture is the first step in designing an information system. While TOGAF, for example, in its Phase C makes it possible to decide whether application or data architecture is to be designed first, in the IHC methodology the database complexity is the main parameter of complexity; therefore, minimizing the number of data storage elements is its primary objective.

[1] DFD – Data Flow Diagram

Implementation

When a decision has been made in which system the requested change is to be implemented, the solution is set up and implemented, during which phase the complexity may further grow. This we can prevent by adhering to the following principles:

- Every implemented functionality must be essential for the process.
- If the process requires more than the existing IS settings allow, we try to simplify the process first.
- User and management requirements for extending of organization structures, adding process variants or changing user interfaces need to be rejected, if they lead to adding even a single element in the model.

A great majority of requests made during IS implementation system from ignorance; functionalities are often implemented that the users were used to in the old system. If the data model is designed properly in line with the process model, there is no problem adding possible improvements at any time later on.

2.6 IHC Evaluation

To include the aforementioned methodology in the context of the methodologies presently most frequently used, we will use a meta-description of the methodology taken from the publication of A. Buchalcevová called '*Methodologies for Building Information Systems'* (Buchalcevová, 2005), which summarizes its basic properties, as shown in Table 4.

Table 4 – IHC methodology meta-description; source: author

IHC Meta-Description			
Item	**Meaning, Register Values**		
Methodology ID	IHC		
Methodology title	Identify High Complexity		
Acronym	IHC		
Authors	Ilja Holub		
Year of origin	2015		
Focus	EM	Global methodology	
Scope	Phases	GST, IST, UST, GAN, DAN, IMP, ZAV, PUR	
	Dimensions	HW, TECH, DAT, FUN, UI, PRA, ORG, EKO	
	Roles	client, supplier, operator	
Methodology weight	LM	Light	
Solution type	NEW	Develop a new solution (greenfield)	
	INT	Integrate the solution	
	UPG	Develop and extend the solution	
	TYP	Implement a type solution	
	USE	Use the solution	
Domain		ERP, ERP II	
Solution approach	ST	Structured development	

3. Testing the methodology during implementation of SAP ERP

This case study describes two connected projects on which the author acted as solution architect and project manager and in this work applied the described IHC methodology for managing the complexity of the information system.

Both projects took place in an exceptionally short time span with very low costs. The first project, which lasted two months, consisted of dividing up manufacturing enterprise A and its information system among two newly created companies B and C with different owners. The second project, which took place half a year later and lasted half a year, was introducing an SAP system at manufacturing enterprise D, which belonged to the same owner that purchased company C. The individual enterprises and their ownership are shown in **Figure 4**.

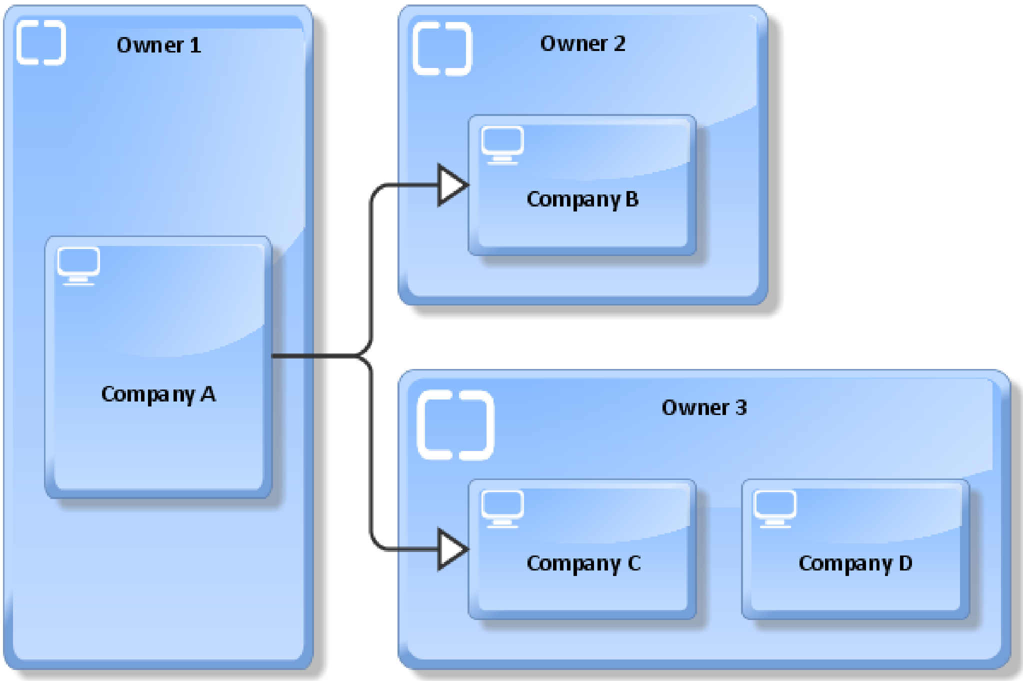

Figure 4 - Division of enterprises among two owners, source: author

3.1 Project 1 – Division of an enterprise

Initial conditions

The initial situation was that there was a manufacturing company A with around three hundred employees using SAP ERP as its main enterprise information system with modules MM, SD, WM, QM, FI, CO, WM and HR communicating with other minor systems and client and supplier systems via EDI and other interfaces. The company's manufacturing program was comprised of several different categories of product. This company found itself in insolvency and was sold to two different parties, each of which purchased a part of the enterprise with the aim of continuing in its existing field, each with a different part of the original product spectrum.

Project objective

The project was ordered by the insolvency administrator conducting the sale of the two parts of the enterprise to its new owners, who demanded two new functioning businesses from the supplier – an external consulting firm – by fixed deadlines. This meant separating out the relevant part of the enterprise for each of the new owners: putting all relevant processes, data, assets and employees and above all the necessary IS/ICT into two newly established companies that were to function independently from the given date. Part of the order was that the transition must take place smoothly without any production downtime or impact on clients and suppliers. Another factor that made the situation considerably more difficult was the dates demanded for having the new companies up and running, the first of which was six weeks after the project was awarded and the second four weeks after – in both cases on the first of the month during the weekend, which was to be used so that it would not be necessary to stop production on a work day.

Course of the project

In light of the fact that the author of this article worked on the project as project manager for the supplier and the contracting party did not prescribe any methodology, he used and thoroughly applied the IHC methodology described in this work. The first step was thus selecting the relevant dimensions and metrics therefor, as shown in **Table 5**.

Table 5 - Selection of dimensions for Project 1, source: author

Dimension	Model and metric / instrument	Complexity/Relevance
Stakeholders	Stakeholder diagram	Relevant
Procedural	Not available	High/Non-relevant
Data	Not available	High/Relevant
Organisational	IHC_ORG program	High/Relevant
Application	Not available	High/Non-relevant
User interface	Not available	High/Non-relevant
Technological	Component diagram	Medium/Relevant

For each dimension the procedure for measuring it was defined as shown in **Table 6**.

Table 6 - Definition of metrics for dimensions STA, ORG and HW in Project 1

	Stakeholders	Organisational	Technological
Complexity	Team communication	Set-up of organisational structures in SAP system	SAP system interface
Metric	Number of connections	Number of elements	Number of connections
Model	Stakeholder diagram	List of entries in all customisation tables of SAP system organisational structure	Component diagram - systems communicating with SAP
Instrument	Aris Express	IHC_ORG program	Aris Express
Granularity (element)	Team member	Entry in table	System
Granularity (connection)	Non-oriented	Not taken into account	Oriented
Value	**6**	**831**	**9**

The first step was clarifying the objectives, defining the organisational structure, drawing up the project plan and assembling the team. The primary limiting factor was the extremely short duration of the project, which meant maximum simplicity of the steps planned, as well as the broad scope of the project, which placed demands on the qualifications of team members and required knowledge not only of all business processes, from purchasing, logistics, manufacture and sales to accounting, controlling and HR management, but also of all modules of SAP ERP and programming, maintenance and administration of all relevant systems and databases. An overview of the required qualifications is shown in **Table 6**7.

Table 7 - Assignment of sources to individual qualifications in Project 1, source: author

Necessary qualification	Source of coverage
Project manager	Consultant 1
Solution architect	Consultant 1
SAP SD consultant	Consultant 1
SAP MM consultant	Consultant 1
SAP PP consultant	Consultant 2
SAP WM consultant	Consultant 2

Necessary qualification	Source of coverage
SAP QM consultant	Consultant 2
ABAP programmer	Consultant 2
SAP FI consultant	Consultant 3
SAP CO consultant	Consultant 3
SAP Basis consultant	Consultant 4

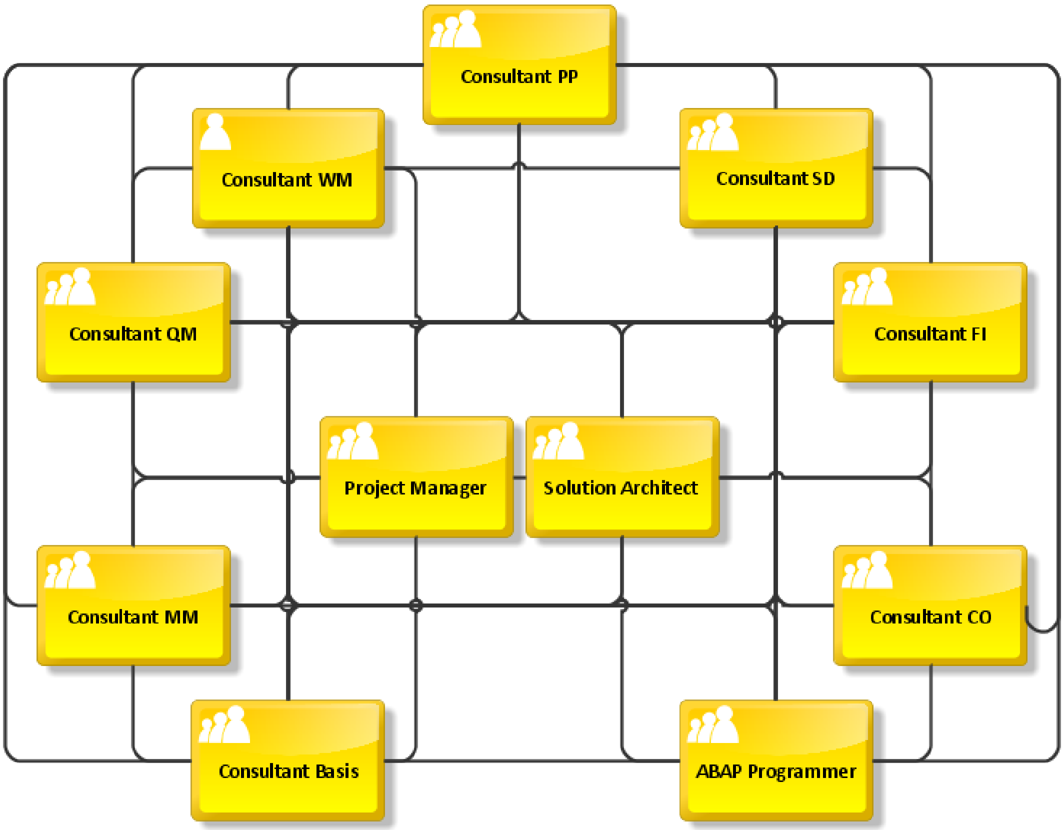

Figure 5 - Diagram of stakeholders dimension before simplification, source: author

Due to the difficulty of managing such a large team, the necessity of individual interfaces to communicate among modules and the consequent time requirements for communication among the consultants, the first step was to simplify this dimension in accordance with the IHC methodology. This was done by engaging specialists who could each cover multiple competencies and thus reduce the complexity of the STA dimension from 55 to 6, i.e. by more than nine fold. A comparison of the original and resulting complexity is shown by the diagrams in **Figure 5** and **Figure 6**.

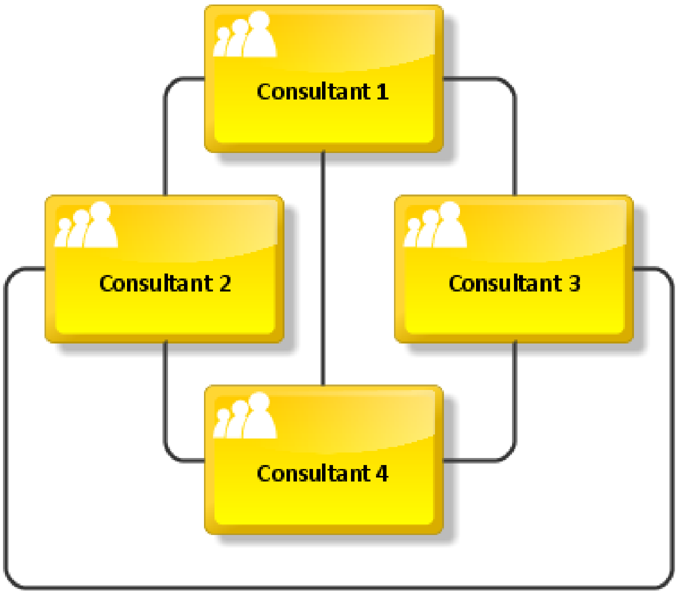

Figure 6 - Diagram of stakeholders dimension after simplification, source: author

The complexity of the procedural dimension could not be measured, as there were no applicable process models and they could not be produced in the given time. This fact contributed to the selection of a solution. The procedural dimension was not addressed at all and the resulting systems were created by copying the existing client into two new clients in the existing SAP system. The complexity of the procedural dimension thus remained the same, but seeing as how no changes were made here, it was possible to not take it into account. The complexity of the data dimension also remained unchanged, as the data objects remained the same, with only their volume reduced – the numbers of master and movement data entries – as each of the new companies received only its own data, with the other company's data removed following the copy. The complexity of the organisational dimension was actually reduced, but in the system this could not be carried out due to its overall complexity, as the master and movement data that could not be de-archived (this was postponed for a later date outside the scope of this project) contained as a key in the database elements of the organisational structure. In order to measure the complexity of the organisational dimension of the original and resulting systems, the IHC_ORG program was used[2].

Another dimension dealt with was the system interface, as it was necessary to separate it for each of the new enterprises. The model thereof is shown in **Figure 7**.

[2] IHC_ORG is a software tool programmed by the author of this article in the ABAP programming language and its output is a number of entries in customisation tables of the given SAP system.

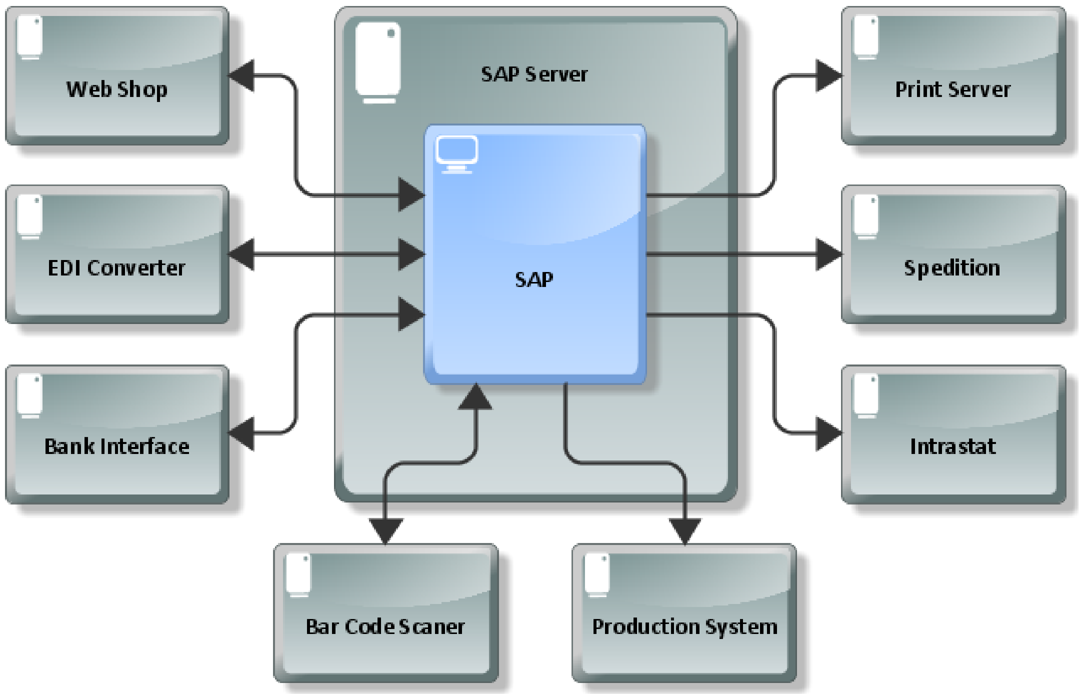

Figure 7 - Model of technological components in Project 1, source: author

Use of the methodology and its benefits

The main benefit of the methodology was dividing the problem up into individual dimensions and dealing with them dependent on their complexity. The STA dimension was considerably simplified and the procedural dimension could be excluded from the project due to its high complexity. These two measures were responsible for the project being successful: it fulfilled the task and met the planned timeframe and cost projection. The secondary objective of the project, reducing the complexity of the organisational dimension at least to half, thus simplifying the whole system and reducing future operating costs, was not met. The reason for this was, for one thing, the high complexity of the organisational dimension, as shown by IHC_ORG, and for another the great volume of data of varying quality that had been input in the system during its more than a decade of operation. It was not possible to accomplish this in such a short time and thus this task was detached into a separate project and moved to a later date. The high complexity of the organisational dimension indicated by IHC_ORG gives rise to the suspicion that it will not be possible to simplify it without changing the procedural dimension.

3.2 Project 2 – Implementation of SAP

Initial situation

The new owner of enterprise C also became the owner of the original company A's entire IT set-up including the SAP system. At the same time, it is also the owner of existing company D – a manufacturing enterprise in the same region that is comparable in size and nature of production to enterprise C. Its information systems are however outdated, insufficiently integrated and do not provide sufficient support for its current business processes.

Project objective

The aim of the project was to make use of the existing ICT infrastructure of enterprise C, in particular its SAP system, for enterprise D as well, i.e. implementing SAP at enterprise D. System administration would remain the responsibility of the IT department at company C, which would charge company D for IT services. In addition to IT, a future aim is to share other processes as well and have some employees, particularly in purchasing and controlling but from other departments as well, work for both companies at the same time.

A limiting factor was, as in the case of the first project, time, namely the wish of the contracting body to implement the system within three months of assignment so that deployment would take place at the start of the calendar year. This time had to suffice for project preparation, analysis, designing and setting up the whole SAP ERP system (in all modules except HR), testing, data migration and going live. The requirement was for the client settings for company C to be copied to the new client company D, potentially adapting any enterprise D processes to the SAP processes functioning at enterprise C. The owner's demand was literally: "processes are identical".

Since enterprise D was comparable to enterprise C in terms of procedural, data and organisational structure, if the IHC methodology had not been applied and the system had been implemented in the same manner as at company C, the overall complexity of information systems C+D owned by the contracting body would have been doubled. The objective of the project manager was however to reduce this complexity in order to reduce the overall project costs and meet the deadline for taking the system live.

Course of project

The author of this article worked on the project as project manager and solution architect and decided to use the IHC methodology. The first step was selecting the relevant dimensions and minimising their complexity, as shown in **Table 8**. As compared to Project 1, the data and procedural dimensions were taken into account. The GUI dimension was addressed only for the custom-programmed screens.

Table 8 - Selection of dimensions for Project 2, source: author

Dimension	**Model and metric / instrument**	**Complexity/Relevance**
Stakeholders	Stakeholder diagram	Relevant
Procedural	Procedure diagram	Relevant
Data	DFD diagram	Relevant
Organisational	IHC_ORG program	Relevant
Applicational	Not available	Non-relevant
User interface	GUI model	Relevant
Technological	Component diagram	Relevant

The structure of stakeholders could not be reduced to the level of the first project, as the capacity of a key consultant could not be secured and had to be replaced with three others.

(**Table 9**) The resulting complexity was expressed by the number 15.

Table 10 shows the same dimensions taken into account in Project 1 and shows that the organisational dimension of enterprise D was markedly reduced, as the overall complexity of both enterprises placed in the shared SAP client C+D grew not by 831, but only by 98, i.e. by 11.8 % compared to **enterprise C.**

Table 9 - Assignment of sources to individual qualifications in Project 2, source: author

Necessary qualification	**Source of coverage**
Project manager	Consultant 1
Solution architect	Consultant 1
SAP SD consultant	Consultant 1
SAP MM consultant	Consultant 1
SAP PP consultant	Consultant 2
SAP WM consultant	Consultant 4
SAP QM consultant	Consultant 5
ABAP programmer	Consultant 5
SAP FI consultant	Consultant 3
SAP CO consultant	Consultant 3
SAP Basis consultant	Consultant 6

Table 10 - Definition of metrics for dimensions STA, ORG and HW in Project 2, source: author

	Stakeholders	**Organisational**	**Technological**
Complexity	Team communication	Set-up of organisational structures in SAP system	SAP system interface
Metric	Number of connections	Number of elements	Number of connections
Model	Stakeholder diagram	List of entries in all customisation tables of SAP system organisational structure	Component diagram - systems communicating with SAP
Instrument	Aris Express	IHC_ORG program	Aris Express
Granularity (element)	Team member	Entry in table	System
Granularity (connection)	Non-oriented	Not taken into account	Oriented
Value	**15**	**929**	**12**

In light of the fact that some processes at enterprise D showed an insufficient degree of maturity, in particular displaying varying variability, it was decided in accordance with the IHC methodology that each such process was to be implemented in the simplest possible variant so as to ensure the desired project outcomes were achieved. An example of the process model is shown in **Figure 8**.

Table 11 - Table 17 - Definition of metrics for dimensions INF, PRO and GUI in Project 2, source: author

	Data	**Procedural**	**GUI**
Complexity	Number of master data repositories	Number of events, activities and printouts (documents) of processes	Number of input fields and GUI functions
Metric	Number of repositories	Number of elements	Number of elements
Model	List of master data objects	Process model	GUI model
Instrument	Excel	Aris Express	Aris Express
Granularity (element)	Object	Event, activity or printout	Field, function key or menu item
Granularity (connection)	Not taken into account	Not taken into account	Not taken into account
Value	**38**	**929**	**14**

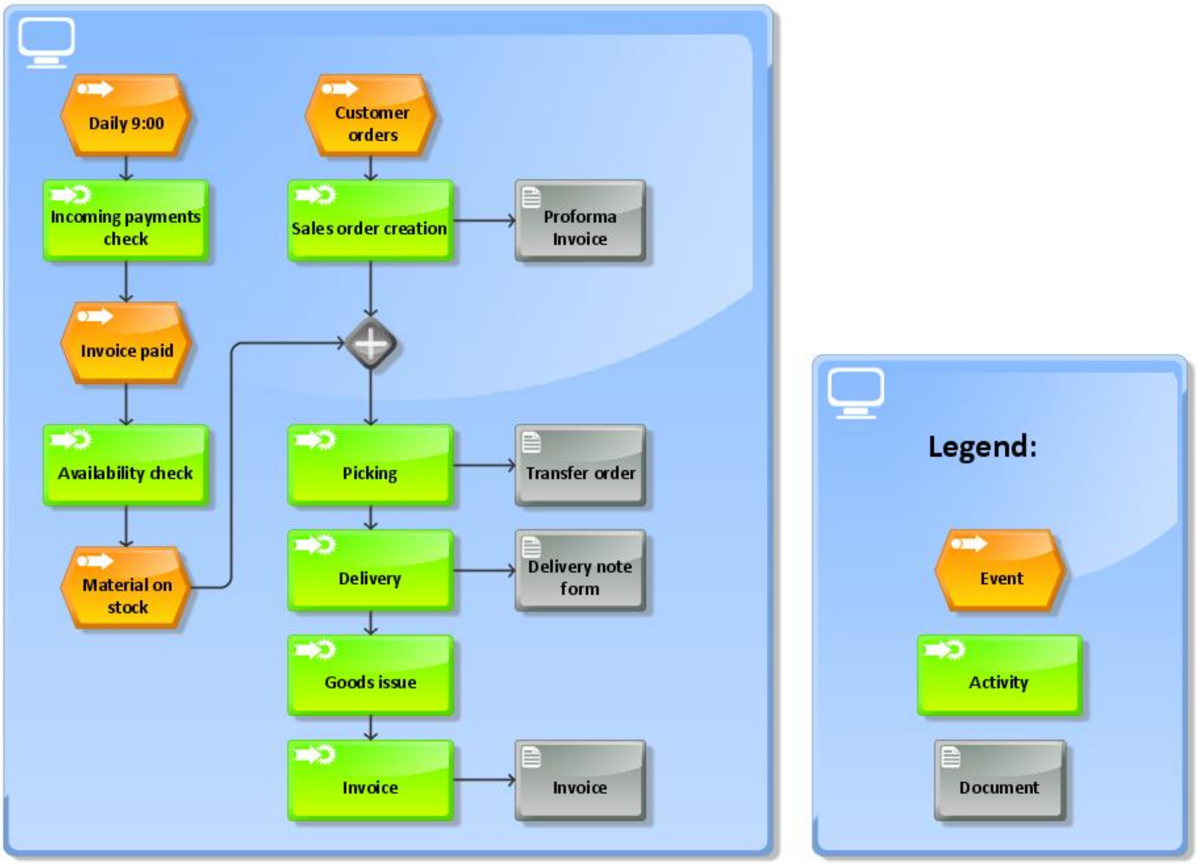

Figure 8 - Model of sales process, source: author

To calculate the data dimension, a list of key data objects was created. This also corresponded to the objects meant for migration from the old systems. Given that both companies have a common owner and some shared clients, suppliers and materials, the variants of shared and separate clients were compared. The list of data objects and repositories thereof are provided in *Table 12 – Variants of data dimension complexity by number of clients, **source: author***.

Table 12 – Variants of data dimension complexity by number of clients, source: author

	Variant 1	Variant 2
Supplier	2	1
Client	2	1
Material	2	1
Chart of accounts	2	1
Quality control regulations	2	1
Forms	2	1

In analysing the data and organisational dimensions, a possibility was found for simplifying them if both companies were integrated into a single client in SAP as two separate company codes and plants. This variant, which did not correspond to the original task but showed lower final complexity, was developed multi-criterially and was shown to provide other benefits as well, including simplification of other dimensions, in particular the procedural dimension. A comparison of the two variants is shown in Figure 9.

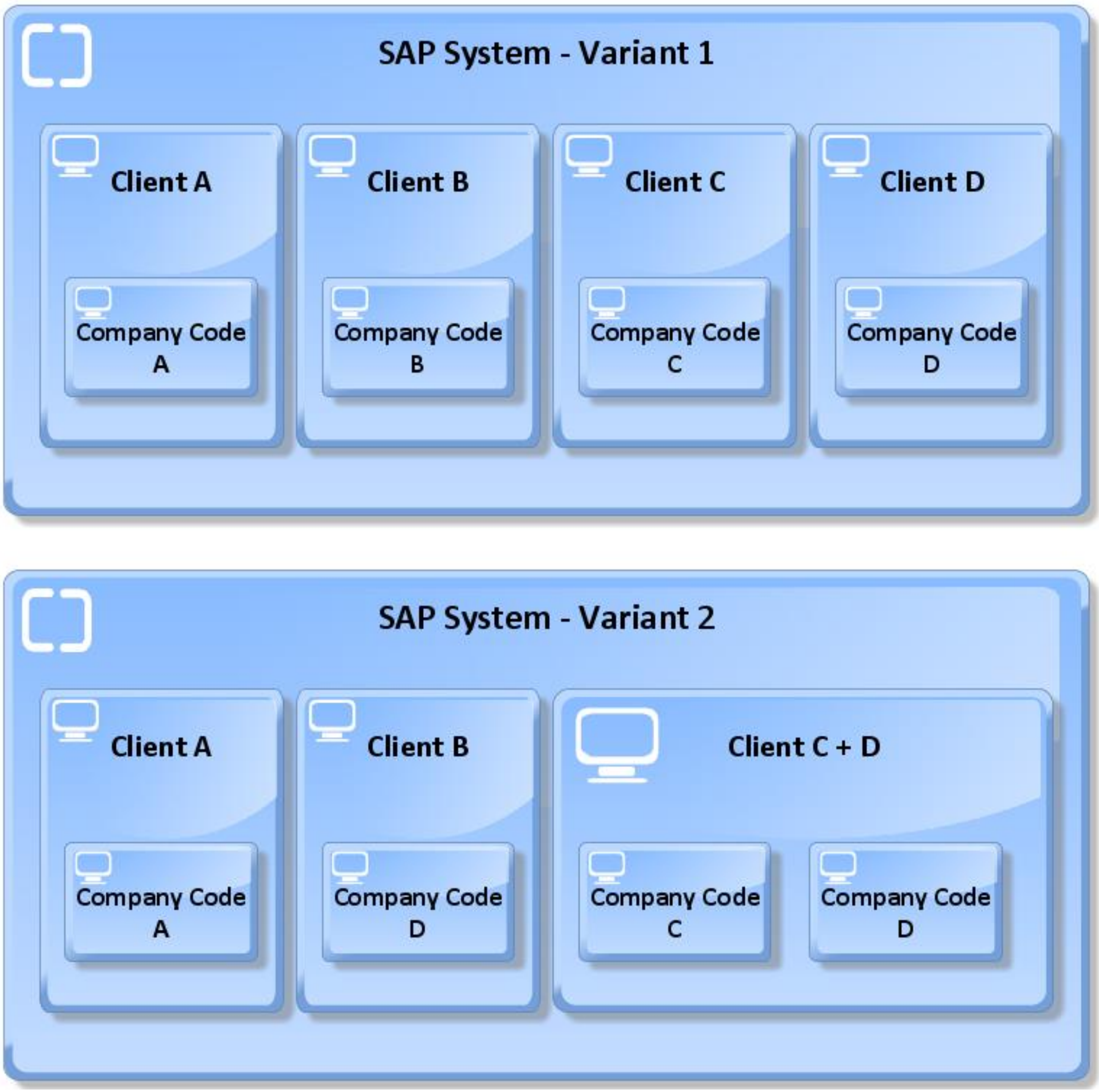

Figure 9 - Design variants for Project 2 - separate or shared client for enterprises C and D, source: author

In addition to the significantly lower complexity of the shared client variant as shown on Figure 10 , other benefits were also shown: Employees working for both companies would not have to log in to two different clients in SAP. In light of the fact that company D purchases certain products from company C, purchasing department employees at enterprise D would be able to see inventory levels at enterprise C and thus plan better. The costs for system operation and maintenance would also be reduced.

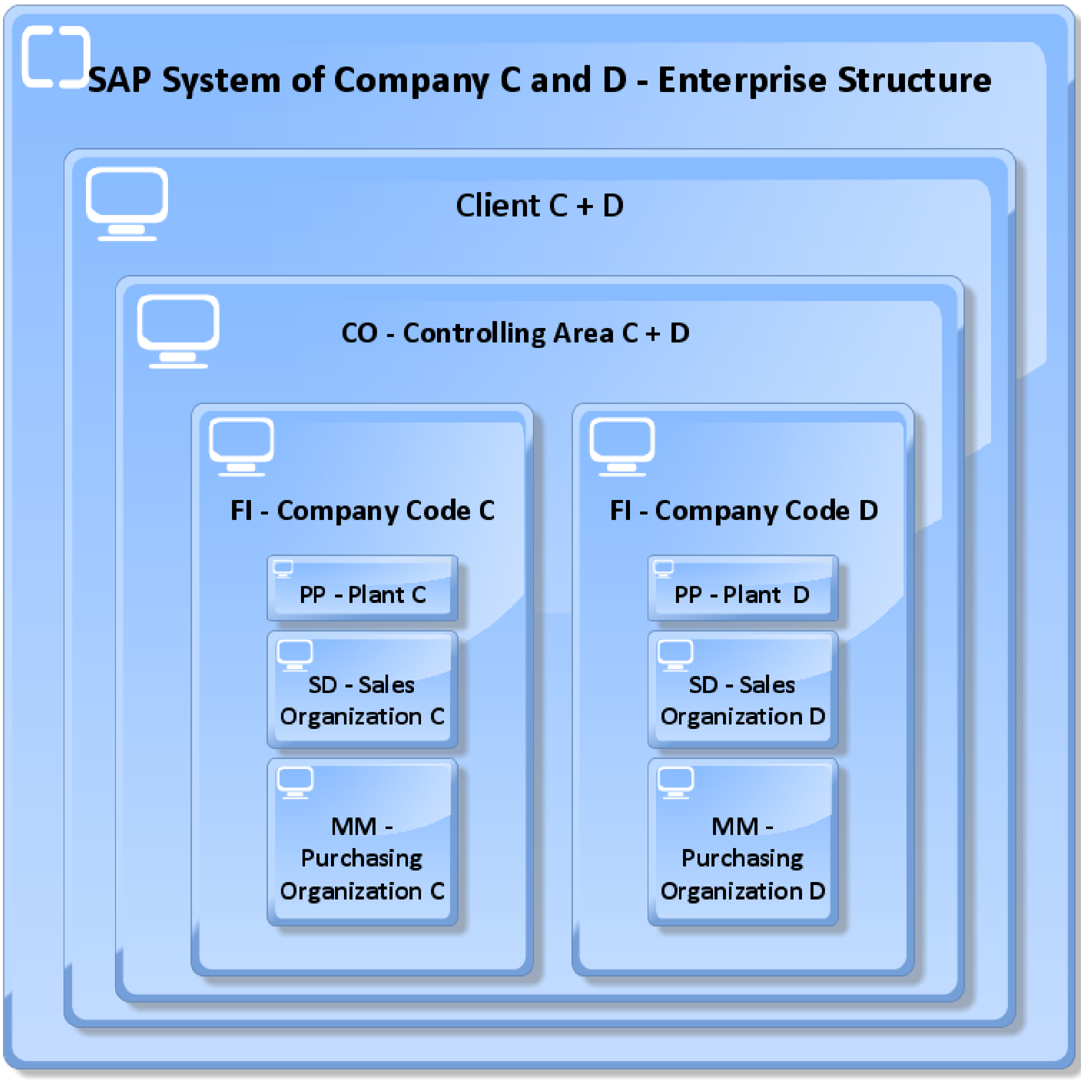

Figure 10 - Organisational structure of both enterprises in SAP, source: author

The advantages and disadvantages of this variant were presented to the contracting body and the variant was approved for implementation. The immaturity of processes involved in having to guess whether and how they would be supported by the system, the workload of key users due to operations and certain fundamental change requests, such as the requirement for implementing barcode scanners for warehouses and the demand for an application to be customised for the key process of a single department, led to a tense situation in November and after the expected exceptional operative task workload of workers during January and February was announced, a decision was made to suspend the project from 15 December until 15 February and move the launch date back to 1 April.

During testing, the advantage of simplifying the system for company C was shown, with the number of necessary testing cases being reduced and the more streamlined organisational structure facilitating testing and data migration. The shifted deadline was met and thanks to the extraordinary efforts of both internal and external project members the project was launched and, though many issues had to be dealt with in the first week, all were resolved on the fly without impacting clients or partners and from the first day the company placed orders, manufactured, supplied and invoiced in SAP.

Despite the fact that the originally planned deadline for deployment was not met, the project was evaluated as a success, because the extension took only a month of real time and in light of the spectrum of implemented modules this was still a very short period compared to comparable projects.

4. Conclusions

The main benefit of IHC for the project was a decision to implement the business system in the same client as Enterprise C. This was made on the basis of measuring the complexity of the data and organization dimensions and selecting a variant with a lower complexity.

Thus, due to applying the IHC methodology, the resulting system complexity could be minimized, which manifested in both easier testing and easier creation of core and flow data during the first weeks of operation, simpler user guides and faster localization and correction of errors. A comparison of the organization dimension complexity using the IHC_ORG program showed that, contrary to the expected 100% rise in complexity, the complexity grew by 12% only, as shown in **Figure 11**.

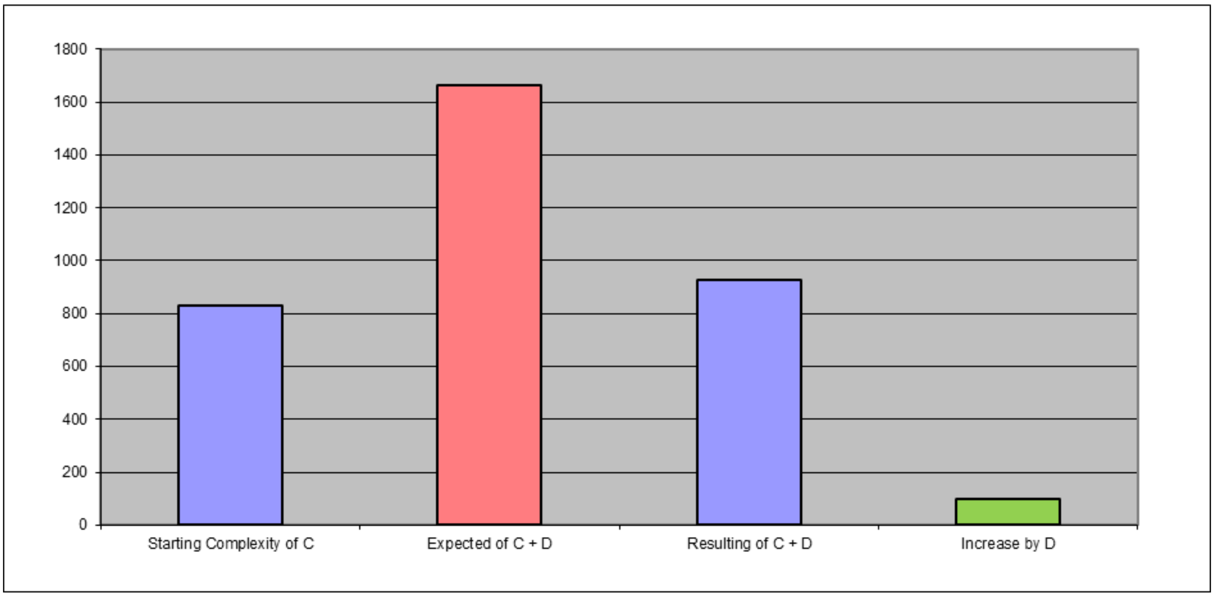

Figure 11 - Organization structure complexity of the SAP system in Enterprises C and D; source: Author

All that while keeping the required system functionality. Based on this outcome, a decision was made to optimize complexity in Enterprise C as well.

Another result of the simplification was a lower number of tested cases, which contributed to a shorter period of implementation.

References

Bansal, V., Negi, T., 2008: A Metric for ERP Complexity, in: *Business Information Systems*. pp. 369–379

Buchalcevová, A., 2005: *Metodiky vývoje a údržby informačních systémů : kategorizace, agilní metodiky, vzory pro návrh metodiky*, 1. vyd. ed. Grada, Praha

Dehmer, M., 2010: *Structural Analysis of Complex Networks*, 1st Edition. Birkhäuser Boston.

Diestel, R., 2010: *Graph Theory*, 4th Edition. Springer

Gartner, 2015: What is Bimodal IT? See Gartner's IT Glossary For The Answer [WWW Document]. Gart. IT Gloss. URL http://www.gartner.com/it-glossary/bimodal/ (accessed 5.23.16).

Geer, D.E., 2009: Complexity Is the Enemy. *Secur. Priv. IEEE* 6, 88.

Holub, I., 2012: Jak a proč vzniká komplexita v IS. *Systémová Integrace*, 19(1): 74-93

Holub, I., 2011. Sborník prací účastníků vědeckého semináře doktorského studia FIS VŠE 2011. Oeconomica VŠE Praha

Holub, I., n.d. Dopad komplexity IS na jejich přidanou hodnotu. Sborník Pr. Účastníků Věd. Semin. Doktorského Stud. FIS VŠE 30–36

Kafura, Henry, 1981: Software structure metrics based on information flow. *IEEE Trans. Softw. Eng.* SE-7

Lemberger, P.P., Morel, M., 2013: *Managing Complexity of Information Systems: The Value of Simplicity,* 1 edition. ed. Wiley-ISTE

Maeda, J., 2006: *The Laws of Simplicity*. MIT Press

Maryška, M., 200:. *Metriky v informatice.* Diplomová práce VŠE, 2006

Mason, P., Cosh, K., 2008: Managing complexity in ICT systems development. *Int. J. Inf. Technol. Manag.* 7, 264–282

Molnár, Z., 2001: *Efektivnost informačních systémů*, 2., rozš.vydání Grada, Praha.

Novotný, O., 2007: Měření výkonnosti podnikové informatiky – teorie a praxe. *CIO Bus. World*.

Phukan, A., Kalava, M., Prabhu, V., 2005: Complexity metrics for manufacturing control architectures based on software and information flow. *Comput. Ind. Eng*. 49, 1–20. doi:10.1016/j.cie.2005.01.005

Repa, V., Bruckner, T., 2015: Methodology for Modeling and Analysis of Business Processes (MMABP). *J. Syst. Integr*.6 (4): 17–28. doi:10.20470/jsi.v6i4.243

Řepa, V., 2006: *Podnikové procesy : procesní řízení a modelování*, 1. vyd. Grada, Praha.

Učeň, P., 2001: *Metriky v informatice : jak objektivně zjistit přínosy informačního systému*, 1. vyd. Grada, Praha.

JEL Classification: M15